(12) UK Patent Application (19) GB (11) 2 385 684 (13) A

(43) Date of A Publication 27.08.2003

(21) Application No 0204242.2

(22) Date of Filing 22.02.2002

(71) Applicant(s)
Sony United Kingdom Limited
(Incorporated in the United Kingdom)
The Heights, Brooklands, WEYBRIDGE,
Surrey, KT13 0XW, United Kingdom

(72) Inventor(s)
Matthew Compton
Stephen Olday

(74) Agent and/or Address for Service
D Young & Co
21 New Fetter Lane, LONDON, EC4A 1DA,
United Kingdom

(51) INT CL[7]
G06F 1/12 , H04L 7/04 12/56

(52) UK CL (Edition V)
G4A AFT
H4K KTKX
H4P PPF

(56) Documents Cited

WO 1999/051032 A1	US 6208665 B1
US 5822317 A	US 5602992 A
US 4761778 A	US 20020003799 A

(58) Field of Search
UK CL (Edition T) G4A AFT, H4K KOT KTKA KTKX, H4P PPF PSEP PSEX PT
INT CL[7] G06F 1/04 1/10 1/12, H04L 7/04 12/56
Other: Online: JAPIO, EPODOC, WPI, IEEEXplore

(54) Abstract Title
Frequency synchronisation of clocks

(57) The frequency of a local clock (30) of a local data processor (4) is synchronised to the frequency of a reference clock (10) of a source data processor (2). Both processors are coupled to an asynchronous switched network (6). The method of synchronising comprises sending, to the local data processor from the source data processor, timing packets each including at least the destination address of the local processor and a field containing reference clock data indicating the time the packet was launched onto the network, and controlling the frequency of the local clock in dependence on the reference clock data.

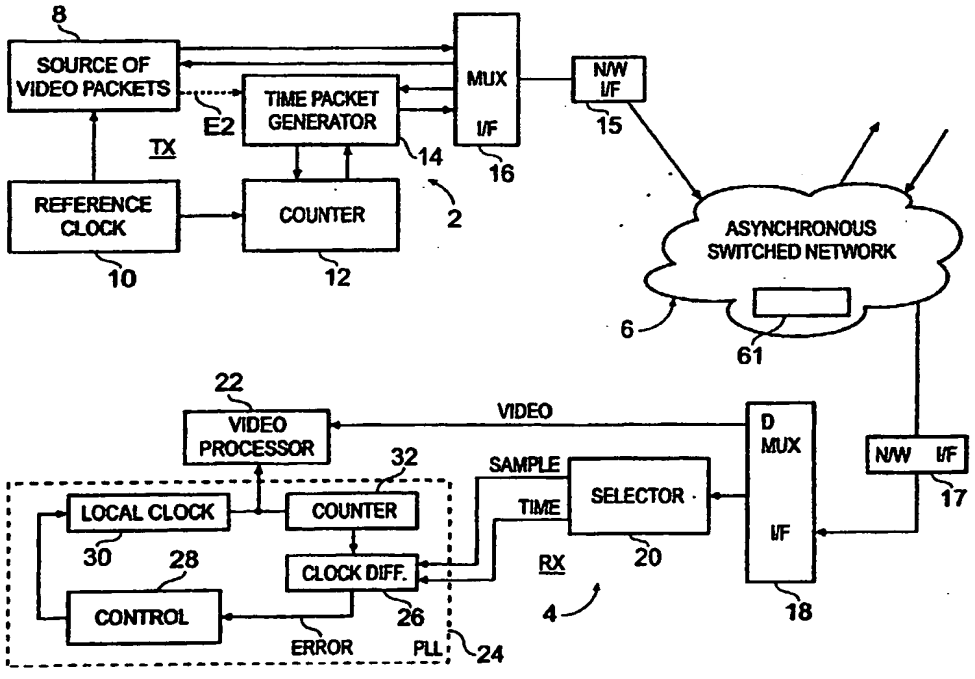


Fig. 1

GB 2 385 684 A

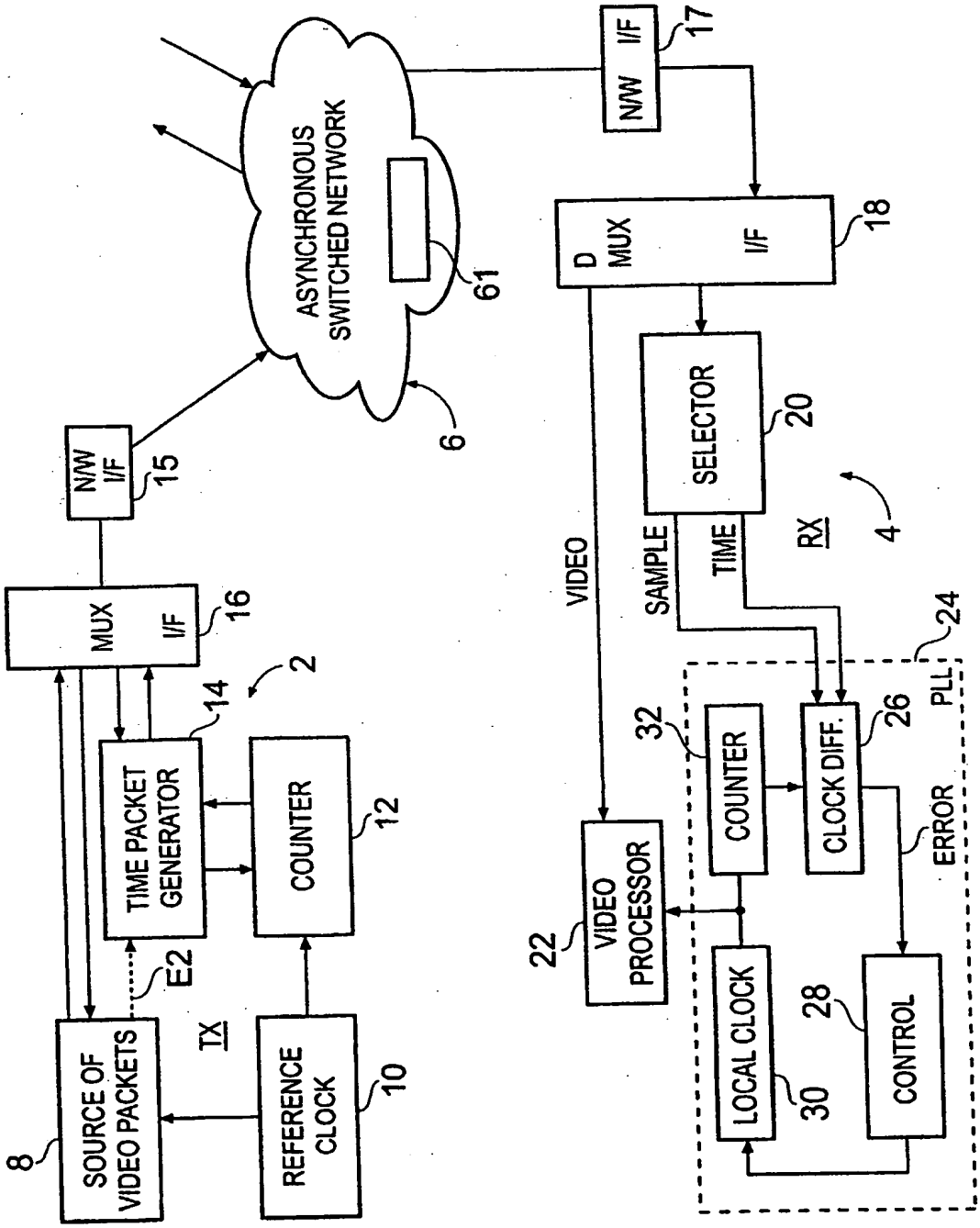
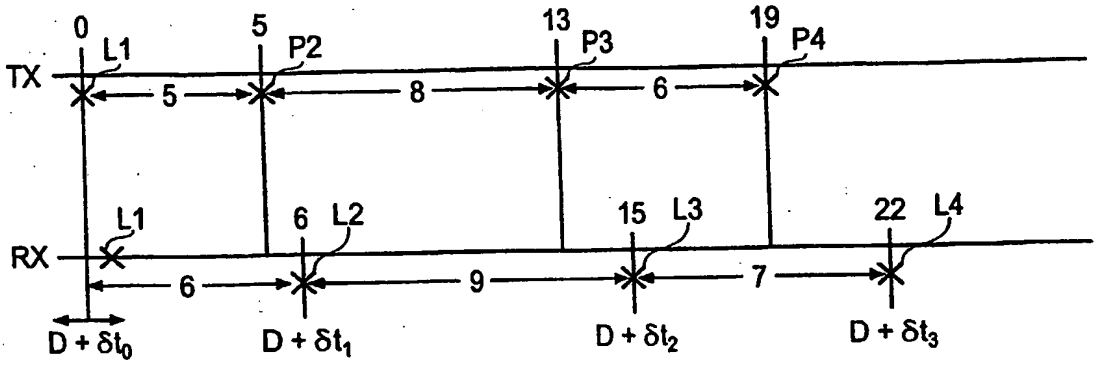

ASYNCHRONOUS SWITCHED NETWORK
61
6
N/W I/F
15
MUX I/F
16
2
TIME PACKET GENERATOR
14
COUNTER
12
E2
TX
SOURCE OF VIDEO PACKETS
8
REFERENCE CLOCK
10
N/W I/F
17
D MUX I/F
18
SELECTOR
20
RX
4
VIDEO
SAMPLE
TIME
24
PLL
32
COUNTER
CLOCK DIFF.
26
ERROR
VIDEO PROCESSOR
22
LOCAL CLOCK
30
CONTROL
28

Fig. 1



Second, RX, diff.	6	9	7
First, TX, diff.	5	8	6
Error	1	1	1
Cumulative Error	1	2	3

Fig. 2

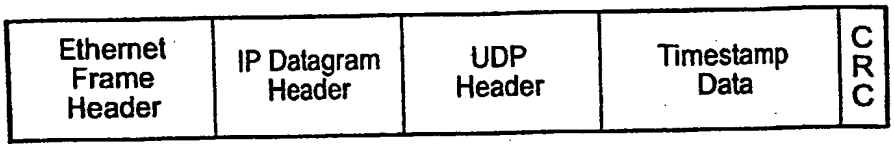


Fig. 3: UDP Timestamp Packet

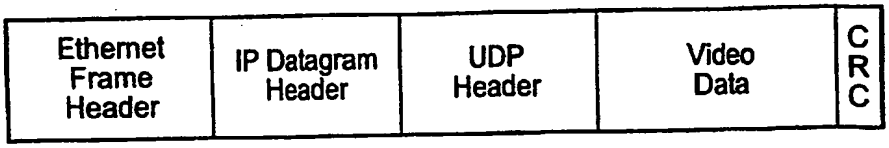


Fig. 7: Video Packet

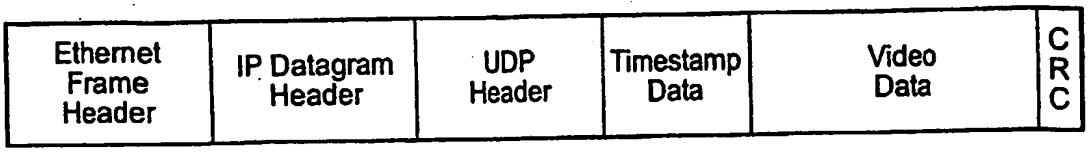


Fig. 8: Combined Packet

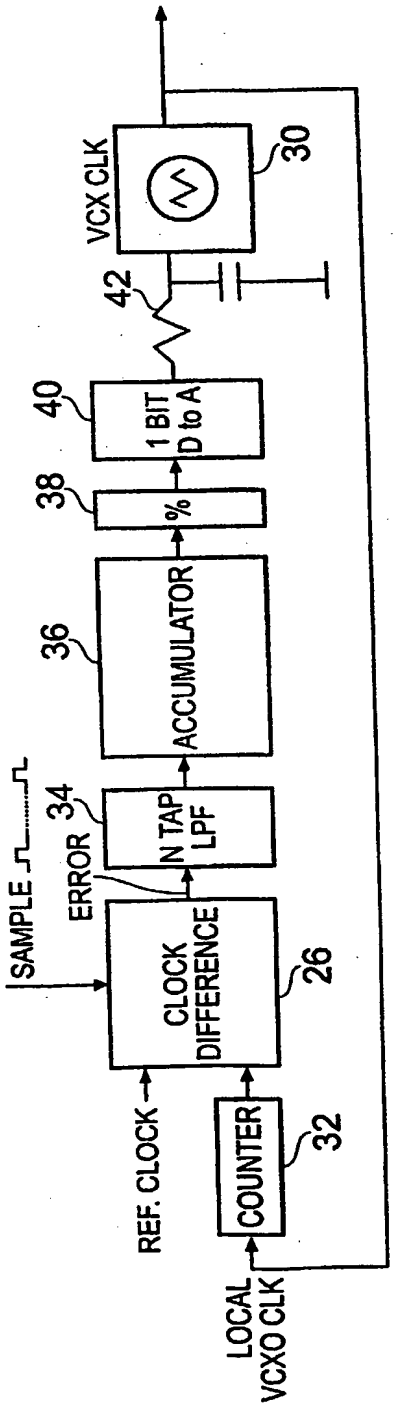


Fig. 4: Frequency Locking System

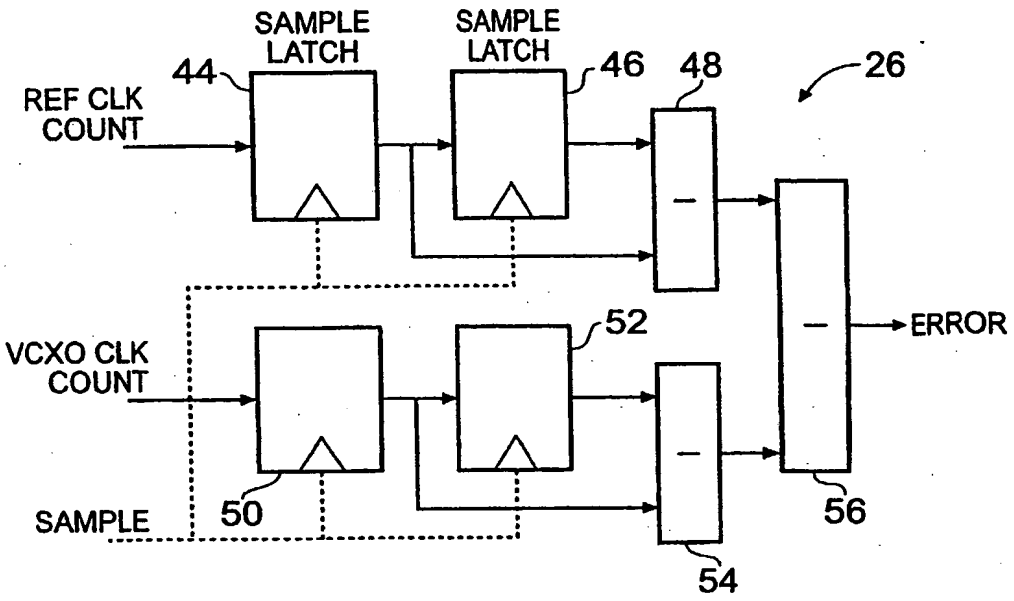


Fig. 5: Clock Difference Circuit

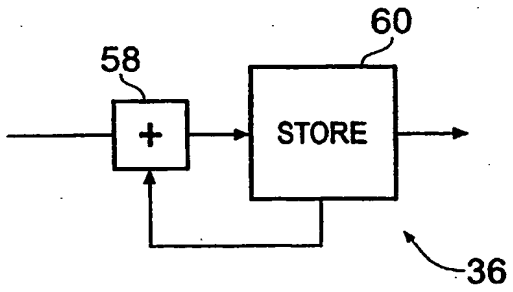


Fig. 6: Accumulator

2385684

Frequency Synchronisation of Clocks

Background of the invention

Field of the invention

The present invention relates to the frequency synchronisation of clocks. Embodiments of the invention relate to frequency synchronisation of clocks over an asynchronous switched network.

Description of the prior art

It has been proposed to distribute, over an asynchronous switched network, data which is generated at a source synchronously with a reference clock. The data may be distributed to many receivers which process the data independently of each other. However the received data needs to be processed synchronously with a local clock in a receiver and that local clock needs to be synchronous with the reference clock. An example of such data is video. There are other examples of such data. An example of such a network is an Ethernet network. There are other examples of such networks.

A prior proposal demonstrated at NAB 2001 distributed video data over a network. Timing data linking local clocks to a reference clock was distributed over another, separate, network

It is desired to provide synchronisation via the network.

ITU-T Rec H222.0 (1995E) discloses that within the ITU-T Rec H222.01 ISO/IEC 13818-1 systems data stream (i.e. MPEG) there are clock reference time stamps called System Clock References (SCR). The SCRs are samples of the System Time Clock (STC). They have a resolution of one part in 27MHz and occur at intervals of upto 100ms in Transport Streams and upto 700ms in Program Streams. Each Program Stream may have a different STC. The SCR field indicates the correct value of the STC of an encoder at the time the SCR is received at a corresponding decoder. With matched encoder and decoder clock frequencies, any correct SCR value can be used to set the instantaneous value of the decoders STC. This condition is true provided there is no discontinuity of timing for example the end of a Program Stream. In practice the free running frequencies of the clocks will not be matched. Thus there is a need to slave the clock of the decoder to that of the encoder using a Phase Locked

Loop (PLL).At the moment each SCR arrives at the decoder it is compared with the STC of the decoder. The difference (SCR-STC) is an error which is applied to a low pass filter and gain stage to generate a control value for the voltage controlled oscillator which is the clock of the decoder.

The system described above uses a synchronous network and locks the absolute time of the decoder clocks to the reference clock.

The present invention seeks to synchronise the frequencies of clocks of data processors linked by an asynchronous packet switched network without requiring infrastructure additional to the network.

SUMMARY OF THE INVENTION

According to one aspect of the present invention, there is provided a method of synchronising the frequency of a local data processor coupled to an asynchronous switched packet network to the frequency of a reference clock of a source data processor also coupled to the network, the method comprising the steps of: sending, to the local data processor from the source data processor across the network, timing packets each including a field containing the destination address of the local processor and a field containing reference clock data indicating the time at which the packet is sent; and controlling the frequency of the local clock in dependence on the reference clock data and the times of arrival of the packets.

Sending timing packets over the network allows the clocks to be synchronised without requiring infrastructure additional to the network. By using, as the reference clock data, data which is that current at the time at which the packet is sent (launched onto the network) the effect of any processing delay or jitter in the source data processor is reduced. For example, a timing packet generator creates a timing packet with an empty time data field. At (or just before) the moment at which the packet is launched onto the network, the reference time is sampled and the time is put into the time data field.

The timing packets may be independent of packets of the data processed by the source data processor and the local data processor. That allows amongst other possibilities all processors on the network to have synchronised clocks whilst allowing data to be sent between selected ones of the processors.

Alternatively, the packets may include both timing data and the said data processed synchronously with the reference clock.

The source data processor may send to the local data processor, across the network, data packets containing at least the address of the local processor and data which is produced synchronously with the reference clock, the timing packets being sent independently of the data packets. Most preferably the source data processor senses when the network has capacity to send the timing data packets and sends them when the said capacity exists.

Preferably, the local clock is controlled in dependence on an error signal which is dependent on the difference between a) the difference between the reference clock data in successively received timing packets and b) the difference between local clock data indicating the local clock time at the times of receipt of the said timing packets. The error signal is updated each time a timing packet is received. Calculating the error signal in this way has the advantage that any discontinuity in the timing packets (for example a packet is not sent or not received) that has no long term effect on the error signal (except that the error signal is not updated for a longer period of time). Also this reduces, if not eliminates the effect of a fixed delay in the processing of the timing packets by the network. Furthermore if the network delay changes then the error signal is affected only at the time of the change and not thereafter.

In an embodiment of the invention, the reference and local clock ticks are counted by reference and local counters. A phase difference between the reference and local counts has no effect on the error signal. These advantages are explained in more detail hereinbelow.

The method may further comprise the step of low pass filtering the said error signal. This reduces the effect of jitter on the error signal. Furthermore the method preferably further comprises the step of accumulating the filtered error signal and controlling the local clock in dependence on the accumulated error signal. This ensures that when the error signal is zero (i.e. the clocks have equal frequencies) the local clock then does not vary in frequency due to the zero control signal. These advantages are explained in more detail hereinbelow.

According to another aspect of the present invention, there is provided a data packet for use in an asynchronous switched network, the packet including at least

a destination address of a processor including a local clock and a field containing a reference clock data indicating the time at which the packet is launched onto the network.

According to a further aspect of the invention, there is provided a data processor comprising a reference clock, a source of data processed synchronously with the reference clock, a generator of timing packets each of which contain reference clock data and an address field containing address data indicating the destination of the packet, and an interface for sending the timing packets across an asynchronous switched network, the reference clock data indicating the time at which the timing packet is sent across the network.

According to a yet further aspect there is provided a data processor comprising a local clock, an interface for receiving, from an asynchronous switched network, timing packets which contain timing data each indicating a reference clock time at the time the packet was sent across the network and an address field containing address data indicating the address of the data processor, and for passing the timing data to a control system for controlling the frequency of the local clock in dependence on the reference clock data and the times of arrival of the timing packets at the processor.

These and other aspects of the invention are set out in the claims to which attention is directed.

BRIEF DESCRIPTION OF THE DRAWINGS

For a better understanding of the present invention, reference will now be made, by way of example, to the accompanying drawings in which:

Figure 1 is a schematic block diagram of an illustrative asynchronous switched network according to the invention to which are coupled a transmitter which transmits video and associated clock data onto the network and a corresponding receiver;

Figure 2 is a schematic timing diagram illustrating the operation of the network of Figure 1 in respect of timing packets;

Figure 3 is a schematic diagram illustrating an example of a timing packet according to the invention;

Figure 4 is a schematic block diagram of an illustrative frequency locked loop (FLL) used in the receiver of Figure 1;

Figure 5 is a schematic block diagram of an illustrative clock difference circuit used in the FLL of Figure 4;

Figure 6 is a schematic block diagram of an illustrative accumulator used in the FLL of Figure 4;

Figure 7 is a schematic diagram illustrating an example of a video packet; and

Figure 8 is a schematic diagram illustrating an example of a timing and video packet according to the invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

First Example

Referring to Figure 1, in this example video is transmitted from a transmitter 2, at one node of an asynchronous packet switched network 6, over the network to one or more receivers 4 (only one shown) at other nodes thereof. The transmitter 2 and the receivers 4 may be, or form part of, network cards. The network in this example is an Ethernet network but could be any other asynchronous switched packet network for example a Token Ring network. The network 6 includes a switch 61 which duplicates the video (and time packets described below) supplied by the transmitter 2 and distributes it to the receivers 4. The video is produced synchronously with a reference clock 10 by a source 8. In this example the video comprises SDI frames of 1440 video samples per line and 625 lines per frame produced synchronously with a 27MHz sampling clock. To decode the video at the receiver 4, the local clock 30 needs to operate at 27MHz (+/- a small tolerance). Thus the local clock 30 needs to be frequency synchronised with the reference clock 10.

In the example of Figure 1, the video data is transmitted across the network 6 as packets in conventional manner. In accordance with an example of the invention time packets, an example of which is shown in Figure 3, are also produced, separately from the video and also transmitted across the network 6. The video packets and time packets are transmitted in separate channels but are associated one with the other by having the same source and destination addresses.

Referring again to Figure 1, the transmitter 2 comprises the source 8 of video packets 8 and the reference clock 10. The video packets are received by an interface and multiplexer 16 which supplies the packets to a network interface 15 The network interface 15 sends the video packets across the network in conventional manner. A counter 12 counts the clock ticks of the reference clock. A time packet generator 14, which operates under the control of the interface 16 obtains the reference count of the counter 12 at any time when the network has spare capacity to transmit a time packet and places it into the time stamp data field (see Figure 3) of a time packet which is then sent across the network. The time data is the time indicated by the reference clock

at the time the packet is sent. The time packets are produced including reference counts and transmitted to a receiver 4 at frequent, but varying, intervals.

The receiver 4 comprises a network interface 17 corresponding to interface 15, and an interface 18 corresponding to the interface 16, which feeds video packets to a video processor 22 and time packets to a time packet selector 20. The selector 20 extracts the timing data from the timing packet and also supplies a sampling signal indicating the time at which the packet was received by the selector 20. The timing data and sampling signal are supplied to a Frequency Locked Loop (FLL) which includes, and controls, a local clock 30. The FLL is a sample data control system. Details of the clocking of samples through the FLL are omitted because such details are not of relevance to the understanding of the present invention and within the normal skill of FLL designers.

The video processor 22 requires the local clock 30 to operate at the same frequency 27 MHz as the reference clock 10 to correctly process the video. As shown in Figures 1 and 4, the FLL comprises a counter 32 which counts the ticks of the local clock 30 to produce a local count and a clock difference stage 26 which is shown in Figure 5. The clock difference stage 26 forms the difference of first and second differences. The first difference is the difference of the reference counts of successive time packets. The second difference is the difference of the corresponding local counts produced at the time of reception of the reference counts. The clock difference stage is described in more detail below with reference to Figure 5.

By way of explanation, attention is invited to Figure 2. The reference and local clocks are ideally operating at exactly 27MHz. However in practice one or both operates with a (small) frequency error. The local clock must operate at the same frequency (+/- a very small tolerance) as the reference clock. Assume for example that the local clock operates at a slightly higher frequency than the reference clock. The transmitter transmits time packets P1 to P4 at irregular intervals. At least one, and preferably a plurality, of packets are transmitted per wrap interval of the counter 12. For example with a 27MHz clock and a 32 bit counter 12, the wrap interval is 159 seconds and at least one packet is transmitted every 159 seconds. Preferably packets are transmitted more frequently than that for example ten per second. The time packets are described in more detail below. In the example of Figure 2, the packets P1 and P2

are transmitted at times spaced by 5 clock ticks of the reference clock 10. The packets P2 and P3 are spaced by 8 ticks and the packets P3 and P4 are spaced by 6 ticks. The packets are received by the receiver after a network delay; assume that delay is constant D. The local counts at the times of reception of the packets P1 to P4 are L1 to L4. The counts L1 and L2 are spaced by 6 local clock ticks. L2 and L3 by 9 and L3 and L4 by 7. Thus the first differences are 5, 8 and 6 and the second differences are 6, 9 and 7 indicating the local clock is operating at a higher frequency than the reference clock. The difference of the first and second differences is the error which is used by the FLL to control the frequency of the local clock.

Forming the error from the first and second differences has the following advantages. The (fixed) delay D has no effect on the error. The absolute values of the reference and local counts are of no consequence. Furthermore, if a time packet is not received it has little effect. For example assume packet P2 is not received: then count L2 is not produced. However the difference (P3-P1) = (P2-P1) + (P3-P2) and (L3-L1)=(L2-L1)+(L3-L2) so (L3-L1) – (P3-P1) =2 in the example of Figure 2 which is the same as the cumulative error with all the packets received.

The foregoing discussion assumes that D is fixed. D is the processing delay of the network. The processing delay in the switch 61 for example is dependent on the average size of the packets switched by it. Thus D may change for instance due to a change in the size of the time packets, which results in a change in the processing delay in the network. If D changes, then stays at its new value, the change affects the difference of the first and second differences only once at the time it changes.

The foregoing discussion also ignores network jitter δt which affects the timing of reception of the timing packets at the receiver, and thus affects the corresponding local counts L. The jitter δt causes a variation in the differences in the times of arrival of the packets at the decoder 4.The jitter δt is regarded as noise. The FLL as shown in Figure 4 includes a Low Pass Filter 34 which low pass filters the error produced by the clock difference stage 26 to reduce the jitter. The filter is for example an N tap digital filter.

The filter 34 is followed by an accumulator 36. An example of the accumulator is shown in Figure 6 which is described in more detail below. The accumulator continuously accumulates the low pass filtered error. The accumulator is

needed to ensure that once frequency lock occurs and thus the error is zero, then the local clock which is a voltage controlled oscillator 30 has a stable, non-zero control value applied to it to prevent "hunting". By way of explanation assume that the local clock operates at 27MHz + X Hertz with zero control input. In the absence of the accumulator, when lock is achieved at 27MHz, then the error and thus the control input is zero so the clock tends to drift towards operating at 27MHz +X. By providing the accumulator, the accumulated error signal forces the clock to operate at frequency lock and when that is achieved the error into the accumulator becomes zero and thus the accumulated value stays constant but non-zero.

The accumulator is followed by a divider 38 which reduces the sensitivity of the clock to small fluctuations (e.g. due to noise) at the output of the accumulator.

The divider 38 is followed by a digital to analogue converter 40 for producing an analogue control value for the voltage controlled oscillator 30. The converter is preferably a single bit converter followed by an RC stage 42 to remove high order harmonics produced by the converter.

The filters 34 and 42, the accumulator 36 and the divider 38 together define the time constant and loop gain of the FLL. The time constant defines the time taken by the FLL to achieve lock. To minimise that time, it is preferable to use the known technique of varying the Low Pass filter 34 and the loop divider 38 to firstly achieve fast but coarse lock and then fine but slower lock.

Time Packet, Figure 3.

In this first example the time packet contains as data only the time data. The Ethernet packet of Figure 3 comprises an Ethernet frame header, followed in order by an IP datagram header, a UDP header, timestamp data which is the reference count mentioned above, and a CRC. The packet contains as address data at least the destination address of the receiver(s) 4. The packet may contain both the source address of the transmitter 2 and the destination address of the receiver(s) to which the transmitter is transmitting. The packet includes data which identifies it as a time packet. That data may be included in one or more of the headers in known manner.

Various types of address data may be provided depending on different operating modes.

In a point to point operating mode in which one transmitter sends data to one selected receiver, the destination address is an address solely of the selected receiver.

In a one to many operating mode in which one transmitter sends data to a group of many receivers, the destination addresses of all the receivers is included (or if they have a group address, the address of the group is included).

In a one to all operating mode in which data is broadcast from the transmitter to all receivers on the network, the address data is a broadcast address which as recognise as applying to all receivers.

The network switch 61 decodes the address data. In the broadcast and group operating modes, it receives one packet from the transmitter and duplicates that packet for transmission to all the receivers designated by the address data.

Clock Difference Stage 26, Figure 5.

The illustrative clock difference circuit of Figure 5 comprises four data latches 44, 46, 50 and 52. The reference count extracted from the time packet is latched into latch 48 in response to the sampling signal which indicates the time at which the selector 20 received the packet. The sampling signal also causes the latch 50 to latch the local count of the counter 28 of the FLL. The previous contents of the latches 44 and 50 are latched into the subsequent latches 46 and 52 in response to the sampling signal. Thus referring to Figure 2, by way of example, latch 44 may contain count P2, latch 46 may contain count P1, and the latches 50 and 52 contain corresponding counts L2 and L1 respectively. A subtractor 48 forms the difference (i.e the first difference mentioned above) of the reference counts in the latches 44 and 46 e.g. P2-P1. A subtractor 54 forms the difference (i.e. the second difference mentioned above) of the local counts in the latches 50 and 52, e.g. L2-L1. A subtractor 56 forms the difference of the first and second differences. The output of the subtractor 56 is the error which controls the local clock 30.

Accumulator 36, Figure 6

The illustrative accumulator of Figure 6 comprises an adder 58 and a store 60. The adder adds the value of the current error (as processed by the filter 34) to the content of the store 60. The store contains the cumulative error shown in Figure 2.

Preferably (and practically) the maximum value storable in the accumulator 36 is limited but the limit is placed outside the normal operating range of the FLL.

1 Bit D to A converter 40, Figure 4

This may be a simple pulse width modulator or a random dither module. A random dither module requires a shorter RC time constant (42) when operating at the centre of its range.

Video Packets

In the first example, the video packets are transmitted across the network 6 separately from the timing packets. The video packets include timing data to enable the video sequence they represent to be correctly reassembled by the video processor 22 at the receiver 4. As shown in Figure 7, the video packets have the same basic structure as the timing packets. The packet includes data which identifies it as a video packet. That data may be included in one or more of the headers in known manner

Video Processor 22, Figure 1.

The video processor 22 may be any video processor including, for example, a monitor, an editor, a special effects machine, and/or a VTR.

Sending timing packets separately from video packets allows timing packets to be broadcast so that all video processors on the network have local clocks frequency synchronised with the reference clock, but also allows video to be sent point to point.

Second Example

In the first example, the time packets are sent separately from the video packets.

Referring to Figure 8, the time data and the video data may be combined in one packet with common address data. The packet includes headers as described with reference to Figure 3 or 7. The packet includes data which identifies it as a combined time and video packet. That data may be included in one or more of the headers in known manner. The time stamp data field which contains a small amount of data precedes the video data field which contains a much greater amount of data. A video sequence is transmitted using many packets. The time data may be included in only some, but not all, of the packets. The time data may be included in a video packet at frequent, but varying, intervals at times when the network has spare capacity as described above.

Referring to Figure1, the a combined video and time packet is generated in the source 8 but the time stamp data field is empty. The packet is fed to the block 14 via the connection E2 shown by a dashed line. The block 14 fills the time stamp data field with the time stamp data at the moment the combined packet is launched onto the network under the control of the multiplexer 16.

Modifications

Whilst the foregoing describes an example of the invention in relation to an Ethernet network, it may be used in any asynchronous switched network e.g. a Token Ring network. The network 6 may be a wired or wireless network or a combination of both wired and wireless.

Whilst the foregoing describes an example of the invention in relation to video, it may be used with any data which is transmitted across an asynchronous network and requires synchronisation of clocks at the transmitter and the receiver(s).

The transmitter 2 and the receiver 4 may be implemented as hardware. They may alternatively be implemented by software in a suitable data processor or as a mixture of software and hardware. A preferred implementation uses programmable gate arrays. It is envisaged that the present invention includes a computer program which when run on a suitable data processor implements the invention.

It will be appreciated that the embodiment of the present invention described above are concerned with frequency synchronisation of clocks. Phase synchronisation of clocks is a different issue which is not addressed by the embodiments although they may be used in conjunction with a phase synchroniser.

CLAIMS

1 A method of synchronising the frequency of a local clock of a local data processor in communication with an asynchronous switched packet network to the frequency of a reference clock of a source data processor also coupled to the network, the method comprising the steps of: sending, to the local data processor from the source data processor across the network, timing packets each including a field containing the destination address of the local processor and a field containing reference clock data indicating the time at which the packet is launched onto the network; and controlling the frequency of the local clock in dependence on the reference clock data and the times of arrival of the packets.

2. A method according to claim 1, wherein the source data processor sends, to the local data processor across the network, data packets containing at least the address of the local processor and data which is produced synchronously with the reference clock, the timing packets being sent independently of the data packets.

3. A method according to claim 1, wherein the source data processor sends, to the local data processor across the network, data packets containing at least the address of the local processor data which is produced synchronously with the reference clock, and the reference clock data indicating the time at which the packet is sent.

4. A method according to claim 1, 2 or 3, wherein the said data packets contain video data.

5. A method according to claim 1, 2, 3 or 4, wherein the source data processor senses when the network has capacity to send the timing data and sends it when the said capacity exists.

6. A method according to claim 1, 2, 3, 4 or 5, wherein the local clock is controlled in dependence on an error signal which is dependent on the difference

between a) the difference between the reference clock data in successively received timing packets and b) the difference between local clock data indicating the local clock time at the times of receipt of the said timing packets.

7. A method according to claim any one of claims 1 to 6, wherein the source processor comprises a reference counter which counts the ticks of the reference clock and the said reference clock data is the count in the reference counter at the time at which the packet containing the reference clock data is sent across the network.

8. A method according to claim 7 when dependent on claim 6, wherein the local processor comprises a local counter which counts the ticks of the local clock and the said local clock data is the count in the local counter of the local processor at the time of receipt of a timing packet containing reference clock data.

9. A method according to claim 8 comprising the step of low pass filtering the said error signal.

10. A method according to claim 9, further comprising the step of accumulating the filtered error signal and controlling the local clock in dependence on the accumulated error signal.

11. A data processor comprising a reference clock, a source of data processed synchronously with the reference clock, a generator of timing packets each of which contain reference clock data and an address field containing address data indicating the destination of the packet, and an interface for sending the timing packets across an asynchronous switched network, the reference clock data indicating the time at which the timing packet is launched onto the network.

12. A processor according to claim 11, comprising a reference counter which counts the ticks of the reference clock and the said reference clock data is the count in the counter at the time the timing packet containing the reference clock data is launched onto the network

13. A processor according to claim 11 or 12, wherein the said source of data forms data packets containing at least data processed synchronously with the reference clock and at least address data indicating the destination address of the data packet and the processor is configured to send said timing packets and data packets across the network separately.

14. A processor according to claim 11 or 12, configured to form packets containing at least the data processed synchronously with the reference clock, address data indicating the destination address of the data packet and the reference clock data indicating the time at which the packet is launched onto the network.

15. A processor according to claim 11, 12 13 or 14, comprising means configured to sense when the network has capacity to send the timing data and to cause the processor to send it when the said capacity exists.

16. A data processor comprising a local clock, an interface for receiving, from an asynchronous switched network, timing packets which contain timing data each indicating a reference clock time at the time the packet was launched onto the network and an address field containing address data indicating the address of the data processor, and for passing the timing data to a control system for controlling the frequency of the local clock in dependence on the reference clock data and the times of arrival of the timing packets at the processor.

17. A processor according to claim 16, wherein the control system comprises a counter which counts the ticks of the local clock and the said local clock data is the count in the counter at the time of receipt of a timing packet containing reference clock data.

18. A processor according to claim 16 or 17, wherein the control system controls the local clock in dependence on an error signal which is dependent on the difference between a) the difference between the reference clock data in successively received

timing packets and b) the difference between local clock data indicating the local clock time at the times of receipt of the said timing packets.

19. A processor according to claim 18, further comprising a low pass filter for filtering the said error signal.

20. A processor according to claim 19, further comprising an accumulator for accumulating the filtered error signal and controlling the local clock in dependence on the accumulated error signal.

21. A processor according to any one of claims 16 to 20, comprising a data processing device for processing data synchronously with the local clock and wherein the said interface is configured to receive data from the network and pass it to the data processing device.

22. A processor according to claim 21, configured to receive packets containing both timing data and the said data and to pass the timing data to the control system and the data to the data processor.

23. A processor according to claim 21, configured to receive separate timing packets and data packets and to pass the timing data to the control system and the data to the data processor.

24. A processor according to claim 21, 22 or 23 wherein the said processing device is a video processing device.

25. An asynchronous switched network comprising a plurality of nodes to at least one of which is coupled a data processor according to any one of claims 11 to 15 , and to at least one other of which is connected a data processor according to any one of claims 16 to 24 and linking the said data processors.

26. A network card comprising a data processor according to any one of claims 11 to15.

27. A network card comprising a data processor according to any one of claims 16 to 24.

28. A timing packet for use in an asynchronous switched packet network, the packet comprising a field including a destination address of a processor and a field containing reference clock data indicating the time at which the packet is launched onto the network.

29. A packet according to claim 28 further comprising header-data, and error correction and/or detection data.

30. A packet according to claim 28 or 29, wherein the packet is an Ethernet packet and the header-data includes an Ethernet frame header, an IP datagram header and a UDP header.

31. A packet according to claim 28, 29 or 30 further including data produced synchronously with the reference clock.

32. A data packet substantially as hereinbefore described with reference to Figure 3 or 8.

33. A data processing system substantially as hereinbefore described with reference to Figure 1 optionally as modified by Figure 4 optionally together with Figure 5 and/or 6.

34. A data processor optionally as hereinbefore described with reference to Figure 4 optionally together with Figure 5 and/or 6.

35. An asynchronous network as hereinbefore described with reference to Figure 1 optionally as modified by Figure 4 optionally together with Figure 5 and/or 6.

INVESTOR IN PEOPLE

Application No: GB 0204242.2 **Examiner:** Geoff Western
Claims searched: All **Date of search:** 22 August 2002

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.T): G4A (AFT) ; H4K (KOT, KTKA, KTKX) ; H4P (PPF, PSEP, PSEX, PT)

Int Cl (Ed.7): G06F 1/04, 1/10, 1/12 ; H04L 7/04, 12/56

Other: Online: JAPIO, EPODOC, WPI, IEEEXplore

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
A	WO 1999/051032 A1 (GRUNDIG)	-
A	US 2002003799 A (TOMITA)	-
X	US 6208665 B1 (LOUKIANOV et al) N.b. page 2, Fig 4	1,11,16,28 at least
X	US 5822317 A (SHIBATA) N.b. pages 2-4, Fig 6	1,11,16,28 at least
A	US 5602992 A (DANNEELS)	-
A	US 4761778 A (HUI)	-

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
&	Member of the same patent family	E	Patent document published on or after, but with priority date earlier than, the filing date of this application.